IN THE CLAIMS

1. (currently amended) A distributed processing system comprising:

a host processor including a host communication infrastructure (HCI) configured to provide communication with said host processor;

a plurality of class processors each ~~having~~ comprising an associated private localized read/write memory, an associated protected localized read/write memory, and public localized read/write memory; and

a plurality of application program interface modules each configured to provide an interface between said host communication infrastructure and at least one said class processor, said application program interface modules each defining a programming interface for a respective said class processor, wherein each said class processor responds to selected data messages on said HCI to perform selected computations utilizing said read/write memory.

2. (original) A distributed processing system in accordance with Claim 1 wherein said distributed processing system is integrated onto a single chip substrate.

3. (original) A distributed processing system in accordance with Claim 2 wherein each of said plurality of class processors is configured to perform operations on a selected proper subset of application objects.

4. (previously presented) A distributed processing system in accordance with Claim 3 wherein said processors are configured to reference other class processors, if at all, only through their respective application program interface modules, without reference to data structures operated upon by said other referenced class processors.

5. (currently amended) A distributed processing system in accordance with Claim 2 wherein said plurality of class processors comprise a plurality of classes of class processors, wherein ~~at least one of said class processors has~~ each said class processor comprises an associated protected localized read/write memory accessible only to itself and to at least one other said class processor of the same class.

6. (original) A distributed processing system in accordance with Claim 5 further comprising semi-private busses coupled to said class processors of said same class providing access to said protected localized read/write memory.

7. (original) A distributed processing system in accordance with Claim 5 wherein said plurality of class processors each further comprise a special purpose processor coupled to said private localized read/write memory, and public read/write memory, and said public read/write memory is configured to be addressable both to said host processor via said HCI and to said special purpose processor.

8. (original) A distributed processing system in accordance with Claim 2 wherein said plurality of class processors comprise a plurality of classes of class processors, and said distributed processing system is configured to restrict direct data communication between said class processors to data communication between class processors of the same class.

9. (original) A distributed processing system in accordance with Claim 8 comprising at least a first said class processor and a second said class processor of the same class, said first class processor further comprises a protected localized read/write memory, and said first and second class processor are configured so that said protected localized read/write memory of said first class processor is addressable by said second class processor.

10. (currently amended) A distributed processing system in accordance with Claim 9 wherein ~~at least one~~ said ~~class processor further comprises a~~ public localized read/write memory

and said class processor having said public localized read/write memory ~~is configured~~ are configured so that said public localized read/write memory is addressable by said host processor.

11. (original) A distributed processing system in accordance with Claim 2 wherein said class processors are controlled and activated by said host processor.

12. (original) A distributed processing system in accordance with Claim 11 wherein said class processors are controlled and activated by said host processor exclusively via said application program interface modules.

13. (currently amended) A method for designing a distributed processing system for an application, said method comprising the steps of

partitioning the application into functions and data messages;

configuring a host processor having a host communication infrastructure (HCI) to pass data messages via the HCI to control the application;

configuring a plurality of class processors to compute the functions into which the application is partitioned in response to the data messages;

configuring each class processor with an associated private localized read/write memory, an associated protected localized read/write memory, and public localized read/write memory; and

interconnecting the class processors to the host processor via application program interface modules in a star configuration, the application program interface modules each defining a programming interface for one or more respective class processors.

14. (currently amended) A method in accordance with Claim 13 ~~wherein at least one class processor comprises a private localized read/write memory, and said method further~~

~~comprises~~ further comprising the step of protecting the private localized read/write memory from being read and from being altered by the host processor and the other class processors, except in response to predefined data messages sent to an application program interface module instructing the class processor to execute a function.

15. (original) A method in accordance with Claim 14 further comprising the steps of:

forming the distributed processing system on an integrated circuit chip; and

locating class processors for executing functions most frequently required by the application most physically proximate the host processor on the integrated circuit chip.

16. (original) A method in accordance with Claim 15 ~~wherein at least one class processor comprises a protected read/write memory, and said method further comprises~~ further comprising the steps of:

grouping functions into groups of related functions;

interconnecting a group of class processors for executing a group of related functions, the group of class processors including the class processor having the protected read/write memory; so that the protected read/write memory is accessible to a plurality of the group of class processors; and

protecting the protected read/write memory from being read and from being altered by the host processor and other class processors not in the group of class processors.

17. (original) A method in accordance with Claim 14 wherein partitioning the application into functions and data messages comprises the steps of:

identifying signals as objects and transforms of signals as functions; and

grouping functions into groups of related functions independent of others of the groups of related functions; and

configuring each of the plurality of class processors to compute a group of related functions to reduce communication between class processors and the host processor.

18. (original) A method in accordance with Claim 17 wherein grouping functions into groups of related functions comprises grouping functions into groups of related functions that have independent data structures, and configuring each of the plurality of data structures comprises configuring each of the class processors to have no knowledge of data structures in other class processors and to communicate with other class processors only through their respective application programming interface modules.

19. (original) A method in accordance with claim 13 wherein interconnecting the class processors to the host processor via application program interface modules in a star configuration comprises the steps of interconnecting the class processors to the host processor via at least one member of the group of interconnections consisting of virtual socket interfaces, I/O port data exchange interfaces, memory mapped dual port random access memory (RAM) banks, stacks, and first-in-first out (FIFO) memory.